

assyuku.c の関数でやっていることまとめ

・ countbytes 関数

ファイルを先頭から順に読み込み、バイトごとにその数字がいくつあるか数え、
kosuu にその数を入れる。

・ dohenkan 関数

1. データのノードを作成する

たとえば数字が 0～4 の五種類あり、countbytes 関数で数えた結果が次のような場合

数字	個数
0	100
1	20
2	10
3	30
4	25

これに終了コードの-1（個数は1）を一個加えて 6 個のノードを作成する。

data=-1 kosuu=1	data=0 kosuu=100	data=1 kosuu=20	data=2 kosuu=10	data=3 kosuu=30	data=4 kosuu=25
--------------------	---------------------	--------------------	--------------------	--------------------	--------------------

この際、各ノードを指すポインタ配列をもう一つ作っておき（forhenkanlist）
変換表を作る際にアクセスしやすくする。

2. ノードをソートする

各ノードを kosuu の降順でソートする。

data=0 kosuu=100	data=3 kosuu=30	data=4 kosuu=25	data=1 kosuu=20	data=2 kosuu=10	data=-1 kosuu=1
---------------------	--------------------	--------------------	--------------------	--------------------	--------------------

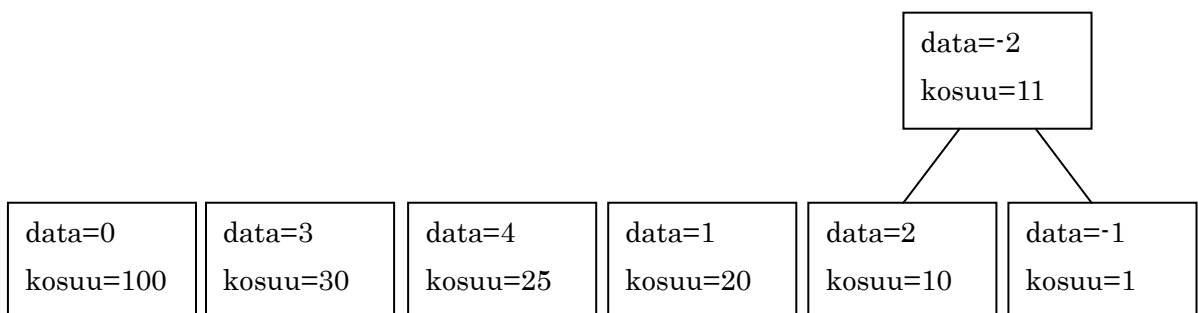
3. ノードを変換ツリーにする

新しいノードを作り、まだ続きがあるということで data=-2 にする。

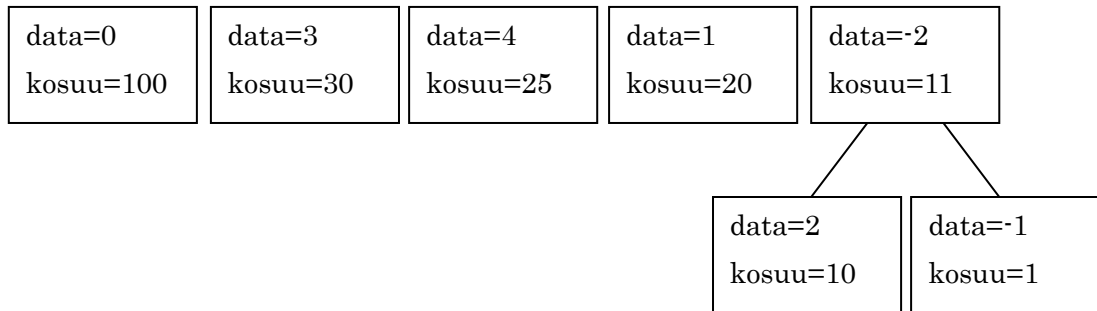
そのノードの next[0] と next[1] に、今あるノードの最後の二つをつなげる。

図では左側が next[0]、右側が next[1] である。

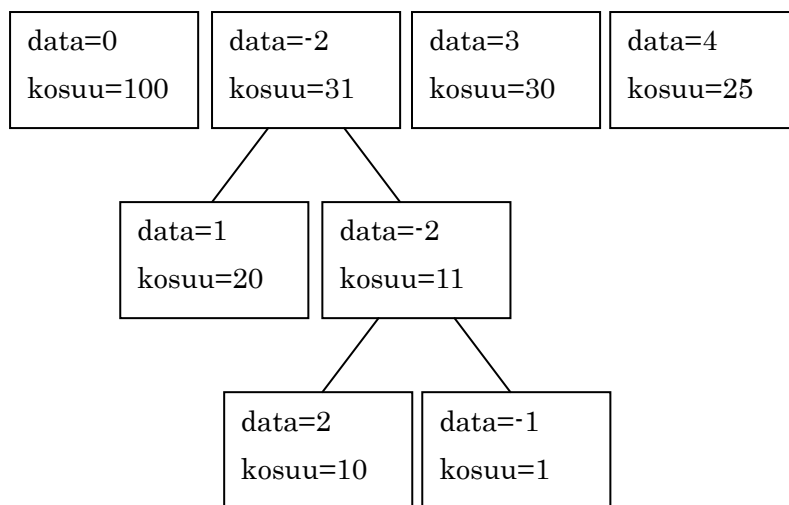
そのノードの kosuu は、つなげた二つのノードの kosuu の合計にする。



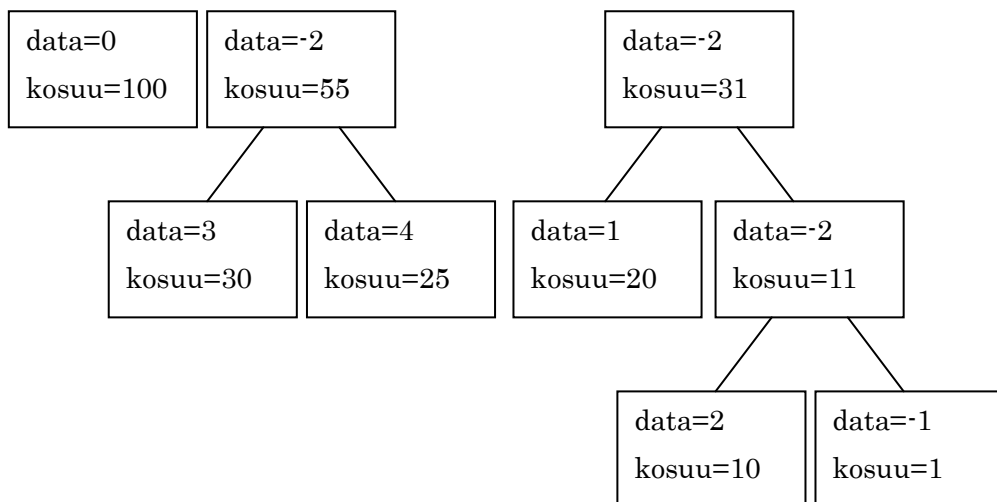
そのノードを sublist に、kosuu が降順になる位置に挿入する。



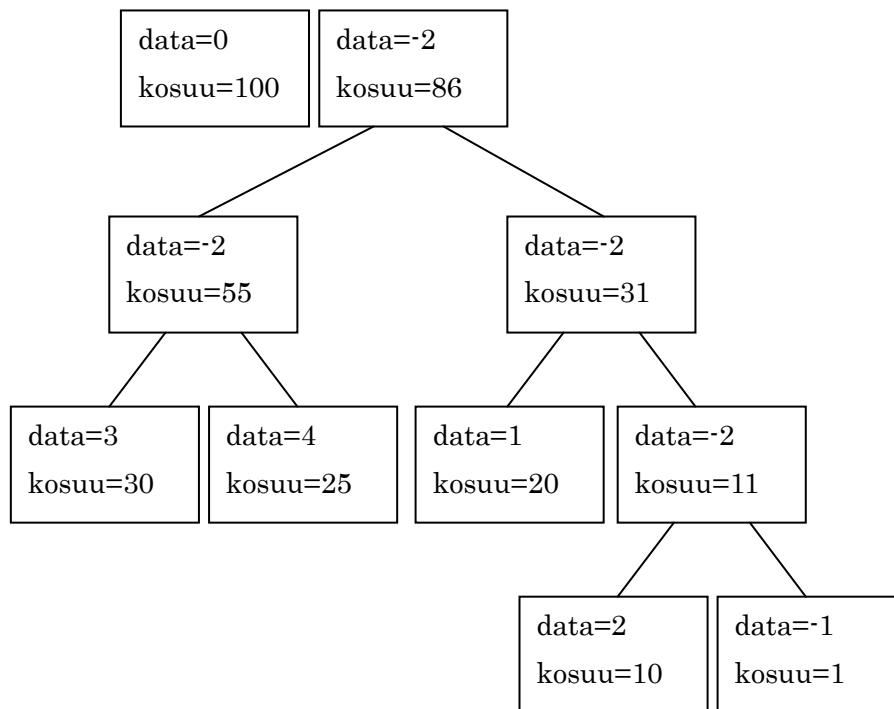
これを繰り返し、sublist に一つしかノードがなくなるまで続ける。



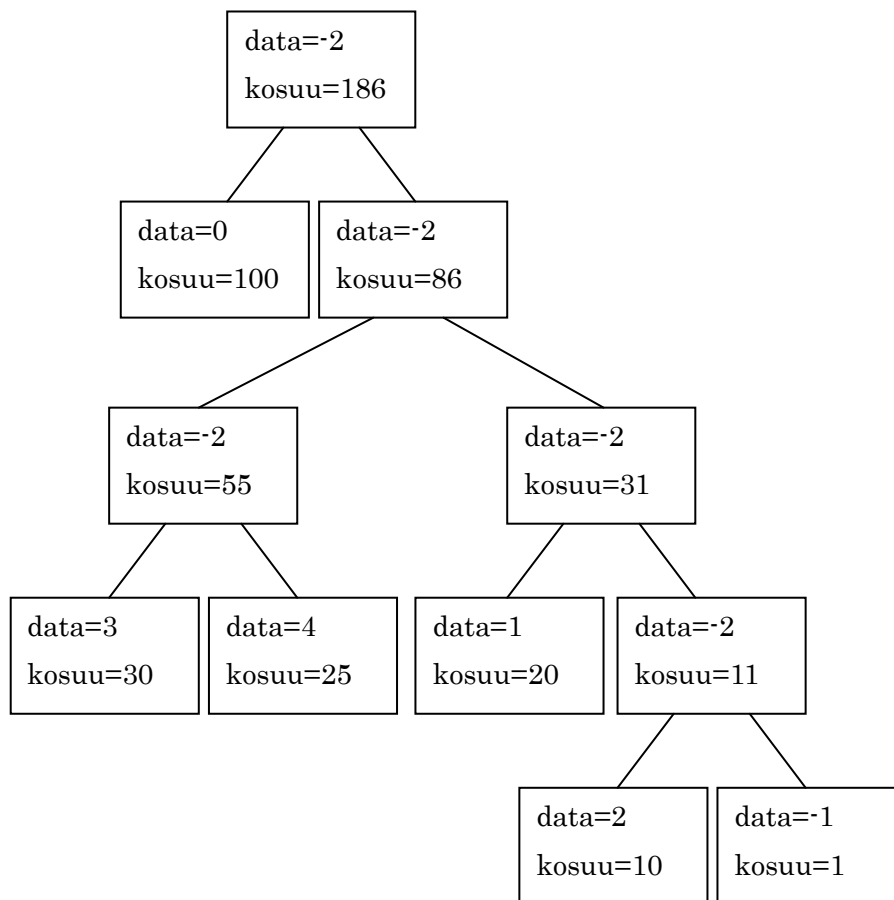
続ける。



続ける。



続ける。これで完成。



4. 変換ツリーを元に、変換表を作成する。

`forhenkanlist` のポインタ情報をもとに、変換ツリーを下からたどっていく。

今のノードが今のノードの `prev` の `next[0]` と同じだったら 0、

`next[1]` と同じだったら 1 を変換表に書き込む。

実際に使うときはこの変換表を逆順に使う。

結果

データ	変換表 (逆順)
-1	1111
0	0
1	110
2	1110
3	100
4	101

5. 変換ツリーの先頭を返す。

この変換ツリーの手元は、解凍の時と、変換ツリーを解放する時に使われる。

圧縮の際は、変換表の方を使う。

- `freehenkan` 関数

再帰を使い、変換ツリーのノードの `next[0]` と `next[1]` を解放した後、ノードを解放する。

- `writetosuu` 関数

MIDI の可変長整数にヒントを得た形式で各データの個数を書き込む。

このデータは解凍時に使われる。

とりあえず個数の下位 7 ビットを書き込むが、個数を 7 ビット右にシフトした数が 0 でなければ書き込むデータの 8 ビット目を 1 にして続きがあるという印をつける。
個数を 7 ビット右にシフトした数を新しい個数にして繰り返す。

- `doassyuku` 関数

ファイルのデータを `writeassyuku` 関数に渡し、圧縮データを書き込ませる。

- `writeassyuku` 関数

変換表をもとに圧縮データを作り、1 バイトたまったごとに書き込む。

- `readtosuu` 関数

ファイルの最初にある個数データを読み込む。

- `dokaitou` 関数

圧縮データを読み込んで `writokaitou` 関数に渡し、解凍を行わせる。

- `writokaitou` 関数

圧縮データのビットごとに `next[0]` か `next[1]` へ変換ツリーをたどり、一番下までいったらその `data` を書き込んで最初に戻る。